IN THE SPECIFICATION:

Please amend Page 10, Lines 18-28 to read as follows:

The interactive graphics stream is a graphics stream used for interactive presentation of graphics objects. This completes the description of the AVClip structure. In addition to AVClips that consist of a plurality of elementary streams, as described, there are also ~~elementary streams~~ AVClips that consist of a single elementary stream. An AVClip consisting of a single elementary stream is called a SubClip. A SubClip is a file that stores only an interactive graphics stream, only an audio stream, or only text data. An interactive graphics stream stored as a SubClip is loaded to the memory in advance for synchronous playback with another AVClip. This operation of loading a SubClip to the memory in advance is called "preloading".

Please amend Page 19, Lines 6-20 to read as follows:

FIG. 8 is a view showing the internal structure of an ICS. The ICS contains one interactive_composition structure or a fragment of one interactive_composition structure. As shown in the left part of FIG. 8, the ICS is composed of the following fields: "segment descriptor" showing that the ICS is an ICS; "video_descriptor" showing the width and height of the video plane in pixels and the frame rate of the associated video stream; "composition_descriptor" showing (i) composition_state, which is information showing whether the type of DS to which the ICS belongs is a Normal Case DS, an Acquisition Point DS, an Epoch Start DS, or an effect_sequence, and (ii) ~~composition_state~~ composition_number showing how many composition have been perform with respect to a screen; "interactive_composition_data_fragment" showing that whether the ICS is carrying an entire interactive_composition structure or a fragment of interactive_composition structure.

Please amend Page 25, Lines 11-13 to read as follows:

FIG. 12 shows a comparison of a page constituted from the page information(y) in DSx and a page constituted from the Page information(y) in ~~DSx+1~~ DSx.

Please amend Page 25, Lines 14-24 to read as follows:

On the other hand, the page constituted from the page information(y) in DSx+1 has three buttons (Button A, Button B, Button C) arranged in the following order: Button [[A]] B, Button [[B]] C, Button [[C]] A. The only difference between the two pages is that the buttons change from being arranged in order of Button A, Button B, Button C to being arranged in order of Button B, Button C, Button A. Even if there is only a slight change between the two pages, the value set in the page_version_number field in the page information(y) in DSx+1 is a greater value than that in DSx. Setting the page_version_number is this way enables even a slight change in the page information(y) to be signaled to the playback apparatus.

Please amend Page 26, Line 25 to Page 27, Line 5 to read as follows:

The playback apparatus manufactured in such a manner includes a BD drive 1, a read buffer 2, a demultiplexer (De-MUX) 3, a video decoder 4, a video plane [[11]] 5, a P-graphics decoder 6, a presentation graphics plane [[18]] 7, a combining unit 8, an I-graphics decoder 9, an interactive graphics plane 10, a combining unit 11, an audio decoder 13, a CLUT unit [[19]] 14, a CLUT unit [[20]] 15, a PSR set 16, a system clock 17, a user_timeout timer 18, a selection_timeout timer 19, a composition_timeout timer 20, an effect_duration timer 21, an operation reception unit 22, and a state control unit 23.

Please amend Page 27, Line 18 to Page 28, Line 1 to read as follows:

The video decoder 4 decodes PES packets received from the demultiplexer 3, and writes the resulting uncompressed picture data to the video plane [[11]] 5.

The video plane 11 is a plane for storing uncompressed picture data. The term "plane" refers to a memory area in the playback apparatus for storing pixel data of one screen. If a plurality of planes are provided in the playback apparatus, the playback apparatus can add the data stored in each plane pixel by pixel to produce one video output, whereby a plurality of graphics images are overlaid as one composite image. The resolution of the video plane [[11]] 5 is 1920 × 1080. Picture data is stored in the video plane [[11]] 5 as 16-bit pixel data expressed in YUV values.

Please amend Page 28, Lines 3-16 to read as follows:

The P-Graphics decoder 6 decodes a graphics stream read from the BD-ROM or HD, and writes the resulting uncompressed graphics to the presentation graphics plane [[18]] 7. The decoding of the graphics stream results in subtitles appearing on the screen.

The presentation graphics plane 18 is a memory having an area for storing one screen of uncompressed graphics. The resolution of this plane is 1920 × 1080, and uncompressed graphics are stored in the presentation graphics plane [[10]] 7 as 8-bit pixel data expressed in index color. The uncompressed graphics stored in the presentation graphics plane [[18]] 7 are displayed after the index color is converted using a CLUT (Color Lookup Table).

[0058]

The combining unit 8 combines (i) the uncompressed picture data with the data stored in the presentation graphics plane [[18]] 7.

The I-graphics decoder 9 decodes an interactive graphics stream read from the BD-ROM or HD, and writes the resulting uncompressed graphics in the interactive graphics plane [[10]] 7.

Please amend Page 29, Lines 6-9 to read as follows:

The combining unit 11 combines the data stored in the interactive graphics plane 10 with a composite image (a combination of the uncompressed picture data and the data stored in the presentation graphics plane [[18]] 7) output form the combining unit 8.

Please amend Page 29, Lines 13-17 to read as follows:

The CLUT unit [[19]] 14 converts index colors of the uncompressed graphics stored in the ~~video plane 19~~ presentation graphics plane 7 to Y, Cr, and Cb values.

The CLUT unit [[20]] 15 converts index colors of the uncompressed graphics stored in the interactive graphics plane 10 to the Y, CR, and Cb values.

Please amend Page 30, Lines 12-21 to read as follows:

The selection_timeout timer 19 is set to the value of the ~~selection_time_out_duration~~ selection_time_out_pts field at the time specified by the PTS(DSn[ICS]). Upon the ~~selection_time_out_duration~~ selection_time_out_pts field value being set, the selection_timeout timer 19 starts counting down in accordance with the system clock. The selection_timeout timer 19 is reset to the ~~selection_time_out_duration~~ selection_time_out_pts field value each time a user operation is performed. If no user operation is received before the selection_timeout timer 19 reaches zero, the selection_timeout timer 19 times out. The selection_timeout timer 19 timing out indicates that the valid interactive period shown in FIG. 9 has ended.

Please amend Page 34, Lines 3-11 as follows:

The processing in step S101 to step S116 having the loop structure repeats the following as long as reading of functional segments continues: judging whether the ignore flag is set to "1" or not (step S102), if the ignore flag is set to "[[0]] 1", transferring the target functional segment from the coded data buffer 33 to either the composition buffer 36 or the stream graphics processor 34 (step S103), and if the ignore flag is set to "0", deleting the target functional segment from the coded data buffer 33 without transferring (step S104).

Please amend Page 38, Lines 5 to Page 39, Line 5 to read as follows:

FIG. 19 illustrates page rendition in normal playback. In DS0, since the page information(0), the page information(1) and the page information(2) are loaded to the composition buffer 36, a menu composed of the page information(0), the page information(1) and the page information(2) can be rendered. On the other hand, when DS1 is reached, the playback apparatus is able to recognize that the contents of the page information(2) have changed by referring to the page_version_number of the page information. Accordingly, the playback apparatus transfers only the page information(2) from the coded data buffer 33 to the composition data buffer 36, and using this to re-render, updates the storage contents of the graphics plane. Arrows up1 and up2 illustrate updating when DS1 is reached, and re-rendering. According to this updating, the contents of the page information(2) stored in the composition buffer 36 change from including two pieces of button information to including three pieces of button information. When re-rendering of the graphics plane is performed based on the storage contents of the composition buffer 36, the buttons on the page corresponding to the page information (2) will change from being ~~three~~ two in number to being ~~two~~ three.

Arrows up3 and up4 in FIG. 19 illustrate updating when DS2 is reached, and re-rendering. According to this updating, the contents of the page information(2) stored in the composition buffer 36 change from including three pieces of button information to including two pieces of button information. When re-rendering of the graphics plane is performed based on the storage contents of the composition buffer 36, the buttons on the page corresponding to the page information (2) will change from being [[two]] three in number to being ~~three~~ two.

Please amend Page 43, Lines 25-27 to read as follows:

A "normal_state_info" structure defines the normal state of button(i), and is composed of "normal_start_object_id_ref", "normal_end_object_id_ref", and "~~normal_repeated_flag~~ normal_repeat_flag" fields.

Please amend Page 45, Lines 1-6 to read as follows:

The selected_end_object_id_ref field specifies the last one of serial numbers assigned to the sequence of ODSs used to present the ~~normal~~ selected state of button(i) in animation. If the selected_end_object_id_ref field specifies the same ID value as that of the selected_start_object_id_ref, the static image of a graphics object identified by the ID is presented as the button(i).

Please amend Page 50, Lines 21-28 to read as follows:

The following describes the Composition_Object. FIG. 30 is a closeup of the internal structure of an arbitrary composition_object(i). As shown in FIG. 30, the composition_object(i) is composed of the following fields: "object_id_ref"; "window_id_ref", "object_cropped_flag";

"composition_object_horizontal_position"; "composition_object_vertical_position", and "~~cropping_rectangle_info~~ cropping_rectangle".

Please amend Page 51, Line 1 to Page 52, Line 4 to read as follows:

The cropping_rectangle structure is valid when the object_cropped_flag field is set to "1". In FIG. [[24]] 30, arrows wd2 indicate that the internal structure of cropping_rectangle is excerpted to be shown in detail. As shown in the figure, the cropping_rectangle structure is composed of the following fields: "object_cropping_horizontal_position"; "object_cropping_vertical_position"; "object_cropping_width"; and "object_cropping_height".

Please amend Page 55, Lines 8-21 to read as follows:

The state control unit 23 in the second embodiment sets a value in a PSR showing a current page (PSR 11) and a value to a PSR showing a current button (PSR [[11]] 10) according to instructions from the I-Graphics decoder 9. Such instructions may be given by way of (i) direct addressing or (ii) indirect addressing. In the case of direct addressing, the I-Graphics decoder 9 outputs an immediate value to be set to the PSR set 16. In the case of indirect addressing, the status control unit 23 determines a setting value to be set in the PSR set 16 when there has been a change in the playback apparatus status or in the user preferences. The status controller 23 determines the value by performing a "procedure when playback condition is changed" or "procedure when change is requested". Hereinafter, a description is given of the procedures performed for setting the PSR 11 (current page) and the PSR 10 (current button).

Please amend Page 57, Lines 9-15 to read as follows:

The I-Graphics decoder 9 in the present embodiment has a graphics controller 37 ~~in addition to the graphics controller 37~~. The graphics controller 37 updates the display composition in accordance with changes in the values of the PSR 10 or the PSR 11 in the PSR set 16, the timeout of the timers 18-21, or a user operation received by the operation reception unit 22. The procedures performed by the graphics controller 37 are shown in detail in FIGs. 38-45.

Please amend Page 59, Line 13 to Page 60, Line 9 to read as follows:

In step S88, when the current playback position reaches the time specified by PTS(DSn[ICS]), it is judged whether or not the user_interface_model in the ICS is set to Always-OnU/I (step S89). If the user_interface_model in the ICS is set to Always-OnU/I, the interactive graphics plane 10 is put into the turned-on state, and the page stored in the interactive graphics plane 10 is output to the CLUT unit [[20]] 15 and composited with the moving picture (step S90).

[0131]

If the user_interface_model is set to Pop-upU/I, the graphics controller 37 maintains the turned-off state of the interactive graphics plane 10. In other words, the page stored in the interactive graphics plane 10 is not composited with the moving picture, and not displayed. While this turned OFF state is maintained, the graphics controller 37 waits for a Pop-up_On operation from the user. If a Pop-up_On operation is made by the user (YES in step S91), the page stored in the interactive graphics plane 10 is output to the CLUT unit [[20]] 15 and composited with the moving picture (step S90). This control for having the storage content of

the interactive graphics plane 10 output to the CLUT unit [[20]] 15 and composited with the moving picture is referred to as putting the interactive graphics plane 10 into a "turned-on state". Display of a pop-up menu is achieved by this turning off and turning on. The processing subsequently returns to the loop processing in steps S1-S6. According to the described processing, updated page information is re-presented at the time specified by the PTS(DSn[ICS]).

Please amend Page 62, Lines 3-14 to read as follows:

Once the graphics object (p) is designated in the step S20 or S21, the graphics object (p) is rendered to the interactive graphics plane 10 at the position specified by the button_horizontal_position and button_vertical_position field values (step S22). By repeating the above steps for each button_info structures provided in the current page, among a plurality of graphics objects associated with each button state, the first graphics object is rendered for each button to the interactive graphics plane 10. After repeating the above steps, the CLUT unit [[20]] 14 is set so as to present the current page using the pallet data specified by the pallet_id_ref field value of the current page (step S23). This completes the description of the flowchart shown in FIG. 40.

Please amend Page 70, Line 23 to Page 71, Line 4 to read as follows:

An issue when implementing such updating of a graphics object is that presentation of animation on the page may be inhibited. As described in the second embodiment, when a user performs an operation for moving focus from one button to another on the menu, the state of the relevant buttons changes from the ~~normal~~ actioned state to the selected state or from the selected

state to the normal state. When the state changes from the normal state to the selected state, the ODS specified by the selected_start_object_id_ref through to the ODS specified by the selected_end_object_id_ref are presented successively (ODS 14 to 16 in FIG. 28).

Please amend Page 74, Lines 1-8 to read as follows:

Next, when subsequent DSs are read, updating of the graphics object data stored in the object buffer 35 is prohibited. In other words, since there are no ODSs in a Normal Case DS, the stream graphics processor 14 does not perform decoding when a Normal Case DS is read. Although the stream graphics processor 14 reads ODS to the coded data buffer [[37]] 33 when an Acquisition Point DS is read, these ODSs are destroyed in the coded data buffer 33 without being decoded because they are duplicates of the ODSs in the Epoch Start DS.